

сообщения  
объединенного  
института  
ядерных  
исследований  
дубна

P10-94-493

А.А.Вовенко, Ю.А.Кретов,  
С.В.Семашко, А.Г.Скрипничук

ПРОГРАММНЫЙ КОМПЛЕКС ЕХАТАРЕ  
ДЛЯ ДОСТУПА К УСТРОЙСТВУ  
НАКОПЛЕНИЯ ДАННЫХ ЕХВ-8500  
НА КОМПЬЮТЕРАХ ТИПА ИВМ РС/АТ  
ПОД УПРАВЛЕНИЕМ MS-DOS

1994

Программный комплекс EXATAPE для доступа  
к устройству накопления данных EXB-8500  
на компьютерах типа IBM PC/AT под управлением MS-DOS

При использовании емкого накопителя EXB-8500 в качестве периферийного устройства компьютеров PC/AT возникла необходимость в создании многофункционального программного обеспечения для управления этим устройством. Пакет EXATAPE реализует необходимые функции управления устройством в среде программирования Си и Турбо-Паскаль посредством протокола SCSI. Пакет является легко переносимым на другие компьютерные платформы и работает как в реальном, так и в защищенном режиме процессора i386.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 1994

---

Перевод авторов

Vovenko A.A. et al.

P10-94-493

The EXATAPE Package for Control of EXB-8500  
Cartridge Tape Subsystem on IBM PC/AT under MS-DOS

When using an EXB-8500 high-capacity cartridge tape subsystem at PC/AT as peripheral device, multi-functional software for control of this device is required. EXATAPE package realized required control functions for C and Turbo-Pascal programming environments through SCSI-interface. The package is easy portable to any computer platform. It functions both in real and in protected mode of processor i386.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

# 1 Введение

При решении задач массовой обработки и накопления данных важную роль играют программы обмена с емкими носителями информации. Эти программы должны включать в себя широкий набор операций ввода-вывода и взаимодействия с операционным окружением.

Если учесть то обстоятельство, что операционная система MS-DOS [1] для компьютеров типа IBM PC/AT поддерживает только накопители на гибких и жестких дисках, то станут понятны причины разработки пакета программ для управления накопителями EXB-8500 [2] на магнитных кассетах. Операционная система не предоставляет стандартных средств управления данным классом внешних устройств, в то время как интегрированный комплекс EXHATARE позволяет пользователю осуществить такое управление из программ, написанных на языках программирования Турбо-Паскаль и Си.

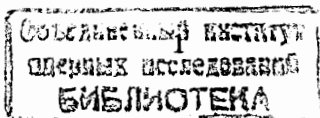
Накопители EXB-8500 используются в тех областях человеческой деятельности (в науке, бизнесе, крупных информационных и вычислительных системах и т.д.), где приходится иметь дело со значительными объемами информации. Так, на одной кассете для накопителя EXB-8500 размещается около 5 гигабайтов (Гб) полезной информации. В физике высоких энергий устройство EXB-8500 также нашло широкое применение (например, в эксперименте ЭКСЧАРМ на У-70, Протвино, Россия).

Пакет EXHATARE предоставляет программисту интерфейс нижнего уровня для управления устройством EXB-8500, при помощи которого можно полностью использовать возможности накопителя. Для создания пользовательского интерфейса высокого уровня с логическим устройством накопления данных и расширенной диагностикой ошибок нужно над пакетом EXHATARE надстроить еще один программный слой, в котором будет скрыта специфика протокола SCSI. Такой интерфейс высокого уровня уже реализован в виде пакета EXAFORT [3] и доступен для использования из программ на языке ФОРТРАН.

## 2 Структура пакета EXHATARE

Программный комплекс EXHATARE состоит из двух частей: резидентного менеджера ввода-вывода EXHATARE.EXE и библиотеки объектных модулей EXHATARE.LIB.

Поддерживаются несколько версий библиотеки объектных модулей для различных компиляторов языка Си: NDP (версии 2 и 3), GNU и TURBO/BORLAND. Кроме того, осуществляется поддержка TPU-модуля для компилятора Турбо-Паскаль (версия 6.0 и выше). Функциональные



возможности TPU-модуля и версий библиотеки полностью идентичны [4], поэтому в дальнейшем мы будем придерживаться единого термина "библиотека EXATAPE".

Устройство накопления данных EXB-8500 имеет интерфейс SCSI [5] (Small Computer System Interface), и для его подключения к компьютеру IBM PC/AT требуется соответствующий SCSI-адаптер.

Менеджер ввода-вывода EXATAPE.EXE представляет собой резидентную программу, которая берет на себя функции взаимодействия как с операционной системой MS-DOS, так и с загружаемыми драйверами, которые требуются для управления дополнительными аппаратными средствами компьютера (например, SCSI-адаптером).

Версия 2.2 резидента EXATAPE.EXE рассчитана на взаимодействие с адаптером SCSI фирмы Adaptec. Поэтому в файле конфигурации CONFIG.SYS операционной системы MS-DOS должна присутствовать следующая строка:

```
DEVICE=C:\SYS\ASPIxDOS.SYS ,
```

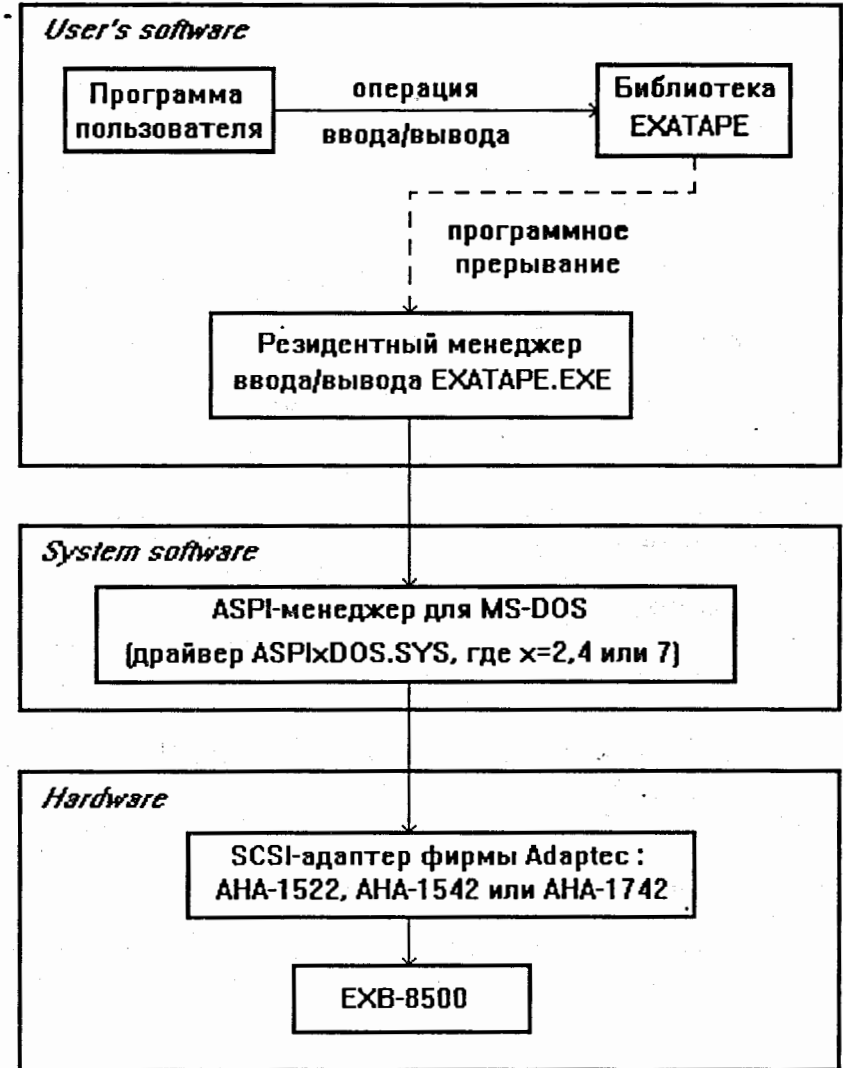
где ASPIxDOS.SYS (x=2,4 или 7) — ASPI-менеджер для MS-DOS, рассчитанный соответственно на один из SCSI-адаптеров: АНА-1522, АНА-1542 или АНА-1742.

При подключении устройства накопления данных EXB-8500 через адаптер АНА-1542 [6] вместо менеджера ASPI4DOS.SYS можно использовать драйвер АНАSHELL.SYS. Но при этом необходимо помнить, что драйвер АНАSHELL.SYS имеет существенный недостаток: при загрузке он формирует только один почтовый ящик (mailbox) для поддержания SCSI-протокола адаптером АНА-1542. Это означает, что данный драйвер нельзя использовать для управления EXB-8500 на тех компьютерах, которые предоставляют для совместного использования в локальной сети свои SCSI-диски, подключенные вместе с EXB-8500 к одному SCSI-адаптеру АНА-1542. Другое следствие из указанного выше недостатка драйвера АНАSHELL.SYS будет описано в пункте 4.2.2 (замечание 3).

Перед выполнением программ, использующих процедуры и функции библиотеки EXATAPE, нужно загрузить резидент EXATAPE.EXE в оперативную память.

Взаимодействие пользовательской программы с накопителями EXB-8500 и с накопителями на жестких дисках (HDD) изображено на рис. 1.

Рис. 1. Схема взаимодействия программы пользователя с устройством накопления данных EXB-8500



### 3 Менеджер ввода-вывода EXATAPE.EXE

Итак, запуск резидентного менеджера ввода-вывода производится обычным образом: либо из командной строки MS-DOS, либо при загрузке системы из файлов AUTOEXEC.BAT или CONFIG.SYS (предложение INSTALL= для MS-DOS версии 4.0 и выше).

При загрузке резидента необходимо указать параметр — номер программного прерывания, которое будет использоваться для передачи управления резиденту. Это число должно быть в диапазоне от 60h до 66h (в десятичном представлении от 96 до 102).

*Пример.*

```
AUTOEXEC.BAT:    loadhi exatape.exe $66
или
AUTOEXEC.BAT:    exatape 102
или
Командная строка: c:\> exatape 0x66
или
CONFIG.SYS       INSTALL=c:\sys\exatape.exe 0x66
```

В процессе загрузки программный менеджер EXATAPE.EXE производит поиск MS-DOS-драйвера для установленного SCSI-адаптера. Версия менеджера 2.2 может опознать или драйвер AHASHELL.SYS, или ASPI-менеджер для MS-DOS, поддерживающий протокол ASPI (Advanced Scsi Programming Interface). В зависимости от обнаруженного драйвера формируется код завершения инсталляции резидента.

### 4 Описание библиотеки EXATAPE

Библиотечные функции логически можно разделить на два уровня — нижний и верхний, причем нижний уровень не зависит от верхнего. Такое разделение было произведено с целью повышения мобильности (переносимости) всего пакета в целом.

Если по каким-либо причинам пользователь не может (или не хочет) работать с функциями верхнего уровня, то, используя лишь функции нижнего уровня, он будет в состоянии корректно управлять устройством EXB-8500.

Функции нижнего уровня служат для взаимодействия с резидентным менеджером ввода-вывода EXATAPE.EXE, который скрывает в себе детальный протокол обмена с операционной системой и с аппаратными средствами компьютера. Таким образом, функции верхнего уровня взаимодействуют с менеджером EXATAPE.EXE не напрямую, а через функции нижнего уровня.

Если для подключения устройства EXB-8500 придется использовать SCSI-контроллер другой фирмы (не фирмы Adaptec), то для адаптации пакета EXATAPE к новым условиям работы нужно будет всего лишь модифицировать резидент EXATAPE.EXE с тем, чтобы он мог настраиваться на MS-DOS-драйверы для нового SCSI-адаптера. При этом не нужно будет производить никакой правки или перекомпиляции программ пользователя и библиотеки EXATAPE.

В случае переноса комплекса на другую компьютерную платформу (другой тип компьютеров, другая операционная система и т.д.) понадобится модификация функций нижнего уровня, число которых невелико по сравнению с числом функций верхнего уровня.

#### 4.1 Нижний уровень

Как уже было сказано ранее, функции нижнего уровня обеспечивают взаимодействие с программным резидентным менеджером ввода-вывода EXATAPE.EXE. Перед обращением к любой функции нижнего уровня нужно быть уверенным в том, что глобальные переменные инициализированы должным образом. Достигнуть этого проще всего, выполнив в самом начале обращение к функции инициализации Init\_IO\_Map (1).

Эта функция производит поиск в ОП резидента EXATAPE.EXE (просматривая таблицу векторов прерываний с 60h по 66h в поисках определенной сигнатуры) и устанавливает номер программного прерывания, используемого для обращения к менеджеру.

Другим важным моментом является то, что все указатели, передаваемые в качестве параметров функциям нижнего уровня, должны иметь следующую структуру: первое слово содержит смещение, а второе — сегментный адрес, то есть должны совпадать с указателями MS-DOS типа FAR. В описании специально используется тип DOSFAR, объявленный как `byte far *`.

##### 4.1.1 Описание глобальных переменных

Три глобальные переменные устанавливаются во время инициализации при обращении к функции Init\_IO\_Map:

```
extern int exatape_rc;
extern byte exatape_int;
extern word exatape_ver;
```

Если резидент EXATAPE.EXE загружен в ОП (код возврата функции Init\_IO\_Man равен нулю), то переменная exatape\_rc (код завершения инсталляции резидента) может принимать следующие значения:

- 1 — для доступа к EXB-8500 используется драйвер ANASHELL.SYS, предварительно загруженный через CONFIG.SYS ;
- 2 — для доступа к EXB-8500 используется ASPI-менеджер, предварительно загруженный через CONFIG.SYS .

Переменная exatape\_int содержит номер программного прерывания, по которому функции нижнего уровня будут обращаться к EXATAPE.EXE — программному резидентному менеджеру ввода-вывода.

Некоторым системно-зависимым приложениям может понадобиться номер версии резидентного менеджера ввода-вывода EXATAPE.EXE, который при инициализации заносится в глобальную переменную exatape\_ver. Старший байт содержит номер версии, а младший — номер подверсии.

#### 4.1.2 Описание функций нижнего уровня

Следует заметить, что описанные ниже функции не обращаются напрямую к прерыванию 21h MS-DOS или к драйверу для SCSI-устройства. Вместо этого генерируется программное прерывание с номером exatape\_int, и управление перехватывает резидентный менеджер EXATAPE.EXE. Для передачи входных и выходных данных используются только четыре арифметических регистра: AX, BX, CX и DX, так как произвольное использование сегментных регистров в защищенном режиме процессора i386 запрещено.

Поиск резидентного менеджера EXATAPE.EXE в ОП :

byte Init\_IO\_Man (void) | (1)

Возвращаемое значение:

- 0 — если инициализация прошла успешно;
- 1 — резидент EXATAPE.EXE в памяти не найден.

Инициализация SCSI-запроса :

void ANA\_Request (id, buf, blen, cdbptr, cdblen) | (2)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;

- DOSFAR buf — буфер ввода-вывода (в области DOS);
- long blen — размер буфера ввода-вывода в байтах;
- DOSFAR cdbptr — дальний указатель на блок памяти в области DOS, в котором содержится 6/10-байтный SCSI-запрос;
- byte cdblen — длина SCSI-запроса (допустимые значения 6 и 10).

Ожидание окончания выполнения SCSI-запроса :

byte ANA\_WaitEndOp (id) | (3)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500.

Выходное значение:

- 0 — если запрос выполнен без ошибок;
- ненулевое значение, если в процессе выполнения возникла ошибочная ситуация (см. приложение 1).

*Замечание.*

После вызова функции ANA\_Request (2) корректный доступ к блоку памяти DOS по адресу buf возможен только после окончания передачи данных SCSI-адаптером. Чтобы быть уверенным в окончании передачи данных и для получения кода завершения операции нужно обратиться к функции нижнего уровня ANA\_WaitEndOp (3).

Выделение блока памяти в области MS-DOS :

DOSFAR GetDosMem (para) | (4)

Входные параметры:

- word para — размер блока в 16-байтных параграфах.

Возвращаемое значение:

- дальний DOS-указатель на выделенную область памяти (смещение равно 0). Если сегмент равен 0 (то есть указатель равен NULL), то это означает, что MS-DOS не смогла выделить блок заданного размера.

Освобождение захваченного блока памяти :

`word FreeDosMem (dosptr) | (5)`

Входные параметры:

- DOSFAR \*dosptr — указатель на дальний DOS-указатель на блок памяти, сформированный функцией (4).

*Побочный эффект.*

Значение указателя \*dosptr будет установлено в NULL.

Получение указателя на блок памяти в области MS-DOS :

`byte *MapDosMem (dosptr, size) | (6)`

Входные параметры:

- DOSFAR dosptr — дальний указатель на область DOS-памяти (первое слово содержит сегмент, второе — смещение), сформированный функцией (4);
- long size — размер блока в байтах.

Возвращаемое значение:

Указатель на область памяти dosptr, поддерживаемый компилятором:

- Для программ, работающих в реальном режиме процессора i386, возвращаемый указатель будет равняться входному параметру dosptr;
- В защищенном режиме возвращается указатель на область виртуальной памяти, которая закреплена за областью реальной памяти по адресу dosptr.

## 4.2 Верхний уровень

Функции верхнего уровня отвечают за формирование SCSI-запросов для устройства EXB-8500.

Так же, как и для функций нижнего уровня, перед началом работы нужно произвести инициализацию путем обращения к функции Init\_exatape (7). При этом необходимо помнить, что после завершения работы нужно произвести деинициализацию путем вызова функции End\_exatape (8).

Отметим, что специально производить инициализацию нижнего уровня не нужно, если вы пользуетесь функцией верхнего уровня Init\_exatape, так как это делается автоматически.

### 4.2.1 Инициализация и деинициализация верхнего уровня

Инициализация верхнего уровня :

`byte Init_exatape (void) | (7)`

Возвращаемое значение:

- 0 — если инициализация прошла успешно;
- 1 — резидент EXATAPE.EXE в памяти не найден;
- 2 — не хватило памяти в области DOS для инициализации.

Деинициализация верхнего уровня :

`byte End_exatape (void) | (8)`

Возвращаемое значение:

- 0 — если окончание прошло успешно (освободили занятую память);
- Иначе — код ошибки MS-DOS [1].

### 4.2.2 Обмен данными между буфером EXB-8500 и ОП

Инициализация передачи данных из буфера EXB-8500 в ОП :

`void SCSI.InitRead (id, buf, lbuf, nblk, sili, fixed) | (9)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- DOSFAR buf — дальний указатель на область DOS-памяти, в которую будут пересылаться прочитанные с EXB-8500 данные;
- long lbuf — длина DOS-буфера в байтах;
- long nblk — для ленты, записанной в фиксированном формате, параметр nblk задает количество последовательных блоков, а для ленты переменного формата - длину блока;
- byte sili — игнорировать или нет несоответствие заданной длины блока nblk и длины реально прочитанного блока для ленты с переменным форматом V:

- 0 — при несоответствии длин код возврата функции нижнего уровня ANA\_WaitEndOp (3) будет равен 2 (Check Condition);
  - 1 — игнорировать несоответствие длин (код возврата функции ANA\_WaitEndOp (3) будет равняться нулю);
- byte fixed — параметр, указывающий формат блоков на ленте:
    - 0 — переменный формат;
    - 1 — фиксированный формат (длина блока задается функцией SCSI\_ModeSelect (20)).

Инициализация передачи данных из ОП в буфер EXB-8500:

```
void SCSI_InitWrite (id, buf, lbuf, nblk, fixed) | (10)
```

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- DOSFAR buf — дальний указатель на область DOS-памяти, в которую будут пересылаться прочитанные с EXB-8500 данные;
- long lbuf — длина DOS-буфера в байтах;
- long nblk — для ленты, записанной в фиксированном формате, параметр nblk задает количество последовательных блоков, а для ленты переменного формата - длину блока;
- byte fixed — параметр, указывающий формат блоков на ленте:
  - 0 — переменный формат;
  - 1 — фиксированный формат (длина блока задается функцией SCSI\_ModeSelect(20)).

*Замечание 1.*

SCSI-адаптер ANA-1542 для передачи данных между буфером EXB-8500 и оперативной памятью использует механизм DMA (Direct Memory Access), что позволяет освободить ЦП от передачи данных для выполнения каких-либо полезных действий.

Благодаря этому при копировании значительного объема информации с диска на EXB-8500 (или с EXB-8500 на диск) можно организовать процесс так, что чтение информации с диска и пересылка данных из ОП в

буфер EXB-8500 будут выполняться одновременно. Время ЦП будет затрачиваться лишь на инициализацию передачи данных (без ожидания окончания передачи) и на обмен данными с диском, как это показано на рис. 2.

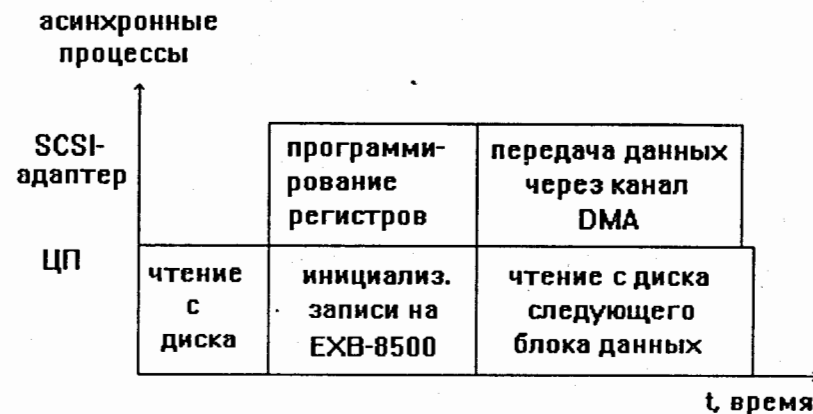


Рис. 2. Параллельная работа ЦП и SCSI-адаптера

*Замечание 2.*

Необходимо помнить о замечании в пункте 4.1.2, согласно которому корректный доступ к блоку памяти DOS по адресу buf для обеих функций SCSI\_InitRead (9) и SCSI\_InitWrite (10) возможен только после окончания передачи данных SCSI-адаптером. Для того, чтобы быть уверенным в окончании передачи данных, и для получения кода завершения операции чтения/записи нужно обратиться к функции нижнего уровня ANA\_WaitEndOp (3) библиотеки EXATAPE.

*Замечание 3.*

Использование драйвера ANASHELL.SYS накладывает ограничение на описанный в замечании 1 алгоритм распараллеливания процессов. Нельзя применять описанную выше технику для передачи информации между EXB-8500 и другим SCSI-устройством (диск, EXB-8500, CD-ROM и так далее), если оба эти SCSI-устройства подключены к одному адаптеру ANA-1542.

#### 4.2.3 Формирование и выполнение SCSI-запросов

В данном пункте описываются многочисленные функции, позволяющие гибко управлять устройством EXB-8500. Внутри каждой функции форми-



руется SCSI-запрос с соответствующим кодом операции и дополнительной информацией для SCSI-адаптера. Инициализация и ожидание завершения запроса реализованы попарно внутри всех функций, описываемых в данном параграфе.

Возвращаемое значение равно 0, если не было ошибок передачи данных; в противном случае все функции возвращают код завершения SCSI-запроса (приложение 1). Для получения более детальной информации о причинах ошибки можно воспользоваться функцией (21), а для получения текстовой строки с диагностикой ошибки — функциями (27)-(29).

Запись блока данных на магнитную ленту EXATAPE реализуется посредством последовательного вызова двух функций: (10) и (3), причем между вызовами может выполняться какая-либо работа, не требующая обращения к области памяти DOS, отведенной под буфер обмена между компьютером и устройством EXB-8500 (для чтения блока данных нужно вызвать сначала функцию (9), затем (3)).

Допустимая позиция ленты для операции записи данных:

- LBOT (логическое начало ленты);
- EOD (признак конца данных);
- FMK (маркер конца файла; при этом FMK перезаписывается);
- блок, следующий за FMK.

Запись на ленту EXATAPE маркеров конца файла (FMK) :

`byte SCSI_WriteFMK (id, nfmk, immed) | (11)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- long nfmk — число маркеров FMK;
- byte immed — параметр, указывающий на необходимость ожидания окончания операции:
  - 0 — ждать окончания операции;
  - 1 — ждать только момента инициализации команды.

Удаление информации с ленты :

`byte SCSI_Erase (id, immed, deldata) | (12)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- byte immed — параметр, указывающий на необходимость ожидания окончания операции:
  - 0 — ждать окончания операции;
  - 1 — ждать только момента инициализации команды;
- byte deldata — параметр, указывающий на необходимость физической перезаписи ленты:
  - 0 — физическое уничтожение данных не нужно (будет записан только признак конца данных EOD);
  - 1 — требуется физическая перезапись информации на ленте.

Сдвиг на заданное число блоков/файлов относительно текущего положения ленты :

`byte SCSI_Space (id, opcode, nblk) | (13)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- byte opcode — код операции:
  - 0 — перемотка на заданное число логических блоков;
  - 1 — перемотка на заданное число файлов (маркеров FMK);
  - 2 — перемотка до признака конца данных (EOD);
- long nblk — количество блоков (или маркеров FMK), которые нужно пропустить при перемотке ленты. Если значение nblk меньше 0, то перемотка ведется по направлению к началу ленты, а если nblk больше 0, то к концу ленты.

*Замечание.*

Если при перемотке на заданное число логических блоков встретится маркер конца файла, то это вызовет аварийное завершение операции. Магнитная головка будет находиться после маркера файла (FMK), если перемотка осуществлялась к концу ленты. В противном случае (перемотка к началу ленты) головка будет стоять на начале маркера FMK.

Позиционирование ленты на блок с абсолютным номером :

byte SCSI\_Locate (id, nblk, immed, blktype) | (14)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- long nblk — номер блока (начиная с 0) относительно начала ленты, под которым нужно установить магнитную головку;
- byte immed — параметр, указывающий на необходимость ожидания окончания операции:
  - 0 — ждать окончания операции;
  - 1 — ждать только момента инициализации команды;
- byte blktype — тип блоков:
  - 0 — логический блок данных или FMK;
  - 1 — физический блок данных (1 K6) устройства EXB-8500.

*Замечание.*

Возможность позиционирования для физических блоков EXB-8500 (blktype равен 1) фирмой-разработчиком пока не реализована.

Перемотка ленты на начало :

byte SCSI\_Rewind (id, immed) | (15)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- byte immed — параметр, указывающий на необходимость ожидания окончания операции:

- 0 — ждать окончания операции;
- 1 — ждать только момента инициализации команды.

Загрузка/разгрузка картриджа для устройства EXB-8500 :

byte SCSI\_Load (id, immed, opcode) | (16)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- byte immed — параметр, указывающий на необходимость ожидания окончания операции:
  - 0 — ждать окончания операции;
  - 1 — ждать только момента инициализации команды;
- byte opcode — код операции:
  - 0 — разгрузка SCSI-устройства (для SCSI-диска данная команда вызывает его остановку, а для EXB-8500 - выгрузку кассеты);
  - 1 — инициализация SCSI-устройства.

Проверка готовности устройства EXB-8500 :

byte SCSI\_TestReady (id) | (17)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500.

Определение текущей позиции ленты :

byte SCSI\_ReadPosition (id, buf, blktype) | (18)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- ReadPosStr \*buf — указатель на структуру ReadPosStr (приложение 3), в которой после выполнения функции будет содержаться информация о текущем положении ленты накопителя EXB-8500;

- byte blktype — тип блоков:
  - 0 — логический блок данных или FMK;
  - 1 — физический блок данных (1 Кб) устройства EXB-8500.

*Замечание.*

Чтение текущей позиции для физических блоков EXB-8500 (blktype равен 1) фирмой-разработчиком пока не реализовано.

Получить параметры настройки устройства EXB-8500:

byte SCSI\_ModeSense (id, buf) (19)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- ModeSenseStr \*buf — указатель на структуру ModeSenseStr (приложение 5), в которой после выполнения функции будут содержаться параметры устройства EXB-8500.

Установить параметры настройки устройства EXB-8500:

byte SCSI\_ModeSelect (id, buf) (20)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- ModeSenseStr \*buf — указатель на структуру ModeSenseStr (приложение 5), которая содержит устанавливаемые параметры устройства EXB-8500.

*Замечание.*

Изменение плотности записи (EXB-8500 или EXB-8200) можно производить, только находясь на начале ленты LBOT. Продолжение записи на уже частично заполненную ленту будет производиться с ранее установленной плотностью.

Получение информации о состоянии SCSI-устройства:

byte SCSI\_RequestSense (id, buf, reset) (21)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- ReqSenseStr \*buf - указатель на структуру ReqSenseStr (приложение 4), в которой после выполнения функции будет содержаться информация о состоянии накопителя EXB-8500;
- byte reset — флаг сброса счетчиков ошибок при операциях чтения или записи:
  - 0 — не сбрасывать содержимое счетчиков;
  - 1 — очистить счетчики после завершения запроса.

Запрет/разрешение на удаление носителя информации:

byte SCSI\_MediumRemoval (id, prevent) (22)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- byte prevent — запрет или разрешение на удаление носителя:
  - 0 — удаление носителя (кассеты EXATAPE) разрешено;
  - 1 — удаление носителя запрещено.

*Замечание.*

После запрета на удаление кассету невозможно извлечь ни при помощи кнопки 'UNLOAD' на панели EXB-8500, ни при помощи команды SCSI\_Load до тех пор, пока не будет получено разрешение на удаление носителя от того же инициатора (пользователя), который запретил удаление.

Запрос информации о SCSI-устройстве:

byte SCSI\_Inquiry (id, buf) (23)

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;

- StdInqStr \*buf — указатель на структуру StdInqStr (приложение 2), в которой после выполнения функции будет содержаться информация об указанном SCSI-устройстве.

Определение верхнего и нижнего пределов размера логического блока данных :

`byte SCSI_ReadBlkLimits (id, buf) | (24)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- BlockLimStr \*buf — указатель на структуру BlockLimStr (приложение 6), в которой после выполнения функции будет содержаться информация о предельных размерах логического блока данных EXB-8500.

Захват SCSI-устройства для уникального использования :

`byte SCSI_Reserve (id, pty3, pty3_id) | (25)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;
- byte pty3 — захват EXB-8500 для:
  - 0 — данного инициатора (SCSI-адаптера, который выдает эту команду);
  - 1 — другого SCSI-устройства с номером pty3\_id.
- byte pty3\_id — номер SCSI-устройства, для которого производится резервирование (если pty3=1).

Освобождение SCSI-устройства :

`byte SCSI_Release (id, pty3, pty3_id) | (26)`

Входные параметры:

- byte id — SCSI-адрес устройства EXB-8500;

- byte pty3 — освобождение EXB-8500 от уникального использования:
  - 0 — данным инициатором (SCSI-адаптером, который выдает эту команду);
  - 1 — другим SCSI-устройством с номером pty3\_id;
- byte scsi\_id — номер SCSI-устройства, для которого было произведено резервирование при помощи функции (25).

#### 4.2.4 Получение текста диагностики ошибки

Статус SCSI-устройства после возникновения ошибки :

`char *SCSI_ErrStat (st) | (27)`

Входные параметры:

- byte st — код завершения SCSI-запроса, возвращаемый функцией (3).

Возвращаемое значение:

- указатель на строку символов, оканчивающуюся символом '\0'. В приложении 1 описаны возможные случаи завершения запросов.

Тип ошибки выполнения SCSI-запроса (для ошибок Check Condition):

`char *SCSI_ErrType (buf) | (28)`

Входные параметры:

- ReqSenseStr \*buf — указатель на структуру (приложение 4), в которой должна находиться информация, возвращаемая функцией (21).

Возвращаемое значение:

- указатель на строку символов, оканчивающуюся символом '\0'. После получения кода завершения, равного 2, нужно сразу же вызвать функцию (21), чтобы сбросить состояние ошибки и иметь более подробную информацию об устройстве. По информации, содержащейся в структуре ReqSenseStr, данная функция возвратит указатель на символьную строку с соответствующей диагностикой типа ошибки.

Например, типы ошибок могут быть следующими:

- устройство не готово;
- ошибка носителя (ленты);
- сбой оборудования;
- неверный запрос;
- смена носителя (ленты);
- попытка записи на заблокированную кассету;
- конец ленты

и так далее.

Подробное сообщение об ошибке  
(для ошибок Check Condition) :

```
char *SCSI.ErrMess (buf) (29)
```

Входные параметры:

- ReqSenseStr \*buf — указатель на структуру (приложение 4), в которой должна находиться информация, возвращаемая функцией (21).

Возвращаемое значение:

- указатель на строку символов, оканчивающуюся символом '\0'. Диагностика выдается по каждому случаю для описанных типов ошибок.

## 5 Экспериментальные данные об EXB-8500

Накопители EXB-8500 обладают встроенным буферным запоминающим устройством емкостью 1 Мб, которое обеспечивает параллельное выполнение процессов передачи данных из ОП в буфер EXB-8500 (из буфера в ОП) и физической записи (чтения) информации на магнитную ленту EXHATARE.

Количество данных на одной кассете (ширина ленты 8 мм, длина — 112 метров), записанной в стримерном режиме (лента движется непрерывно), составляет 4.5-4.8 Гб для различных форматов записи (P6/P5). Под обозначением 1 Гб мы понимаем 1024x1024x1024 байтов.

Скорость физического чтения/записи информации с ленты во внутренних буфер EXB-8500 равна 500 Кб/с.

Если программа пользователя передает данные в устройство накопления EXB-8500 со средней скоростью не менее 500 Кб/с и с небольшим коэффициентом вариации, то EXB-8500, благодаря встроенному буферу, ведет запись на ленту в стримерном режиме. В данном случае критическим субпроцессом является физическая запись информации на ленту, и при заполнении внутреннего буфера EXB-8500 свыше порогового значения прием данных прекращается до тех пор, пока в буфере не останется достаточно свободного места (пороговые значения пользователь может варьировать в некоторых пределах при помощи функции верхнего уровня 20).

В случае, когда скорость потока данных в среднем менее 500 Кб/с, автоматически включается режим СТАРТ/СТОП, при котором лента периодически останавливается. Перед фактической остановкой ленты на нее записываются "пустые" промежутки (GAP). При подготовке данных со скоростью около 10 Кб/с пустые промежутки заняли 15-20% полезной емкости кассеты.

Перемотка полностью записанной ленты занимает около двух минут. Скорость поиска нужного блока (файла) производится примерно в 75 раз быстрее, чем чтение/запись, и составляет около 37 Мб/с.

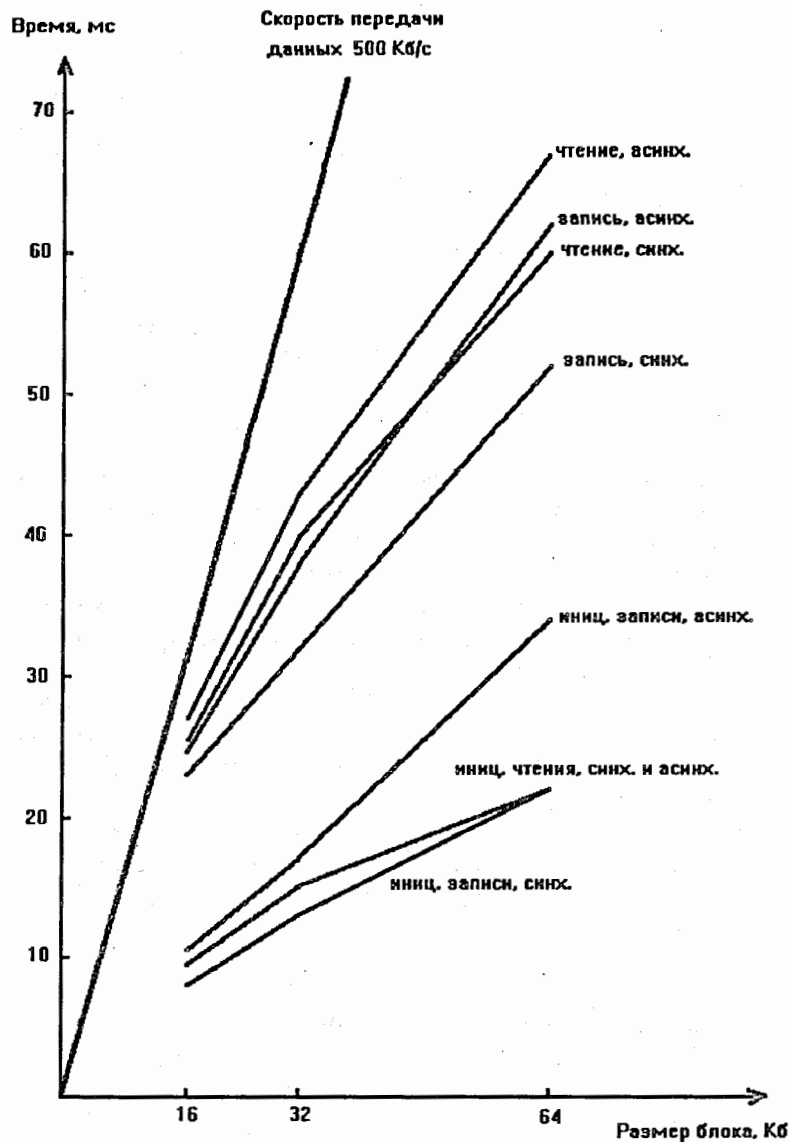
Обмен данными с EXB-8500 ведется логическими блоками, размеры которых могут быть от 1 байта до 240 Кб. Запись информации на ленту может осуществляться блоками как фиксированной, так и переменной длины (для настройки нужно использовать функцию 20).

Авторами экспериментально установлено, что скорость передачи данных из ОП в буфер EXB-8500 (или из буфера устройства в ОП) составляет 1-1.5 Мб/с. На рис. 3 представлены несколько графиков для различных режимов передачи данных.

Термины "синхронная" и "асинхронная" передача данных ("синх." и "асинх.") взяты из [2]; они обозначают режим передачи данных по шине SCSI (максимальная скорость передачи данных равна 4 Мб/с и 1.5 Мб/с соответственно), который устанавливается при помощи переключателя на плате адаптера АНА-1542.

Как уже было сказано выше, адаптер АНА-1542 позволяет во время передачи данных освободить ЦП, используя механизм DMA. При этом использовать ЦП нужно будет только для инициализации передачи блока данных. На рис. 3 графики, помеченные сокращением "иниц.", показывают время, затраченное ЦП на инициализацию передачи данных. Остальные графики показывают время, затраченное ЦП как на инициализацию, так и на ожидание окончания передачи блока данных.

Рис. 3. Время передачи адаптером АНА-1542 данных из ОП в буфер EXB-8500 (запись) и из буфера в ОП (чтение)



Для получения стабильного стримерного режима при копировании большого объема информации с диска на EXB-8500 (или наоборот) необходимо правильно выбрать размер передаваемого блока данных. Если при копировании использовать возможность параллельной работы канала DMA и дискового накопителя, то для достаточно быстрых дисков (скорость чтения/записи около 1 Мб/с) можно обеспечить функционирование EXB-8500 в стримерном режиме, используя даже небольшие блоки (около 16 Кб). Если же не прибегать к столь тонким приемам, как распараллеливание процессов, то размер передаваемого блока данных следует выбрать не менее 32 Кб. Минимальный размер передаваемого блока данных можно получить исходя из скоростных характеристик диска при помощи графиков на рис. 3.

Данная рекомендация относится к блоку данных, передаваемому SCSI-адаптером из ОП в буфер EXB-8500 одной командой, а не к блоку данных на ленте EXHATARE. Если лента записана в фиксированном формате, то адаптер может передавать сразу несколько блоков фиксированной длины (например, 32 блока по 1 Кб).

## 6 Заключение

На базе описанного комплекса EXHATARE авторами были созданы многочисленные программные средства управления устройством EXB-8500. Эти программы активно используются сотрудничеством ЭКСЧАРМ при обработке экспериментальных данных, записанных на магнитных кассетах типа EXAVYTE.

Комплекс EXHATARE имеет слоистую структуру, благодаря которой была повышена мобильность комплекса в целом. Настройка на аппаратные средства PC/AT и драйверы MS-DOS локализована в резидентном менеджере. Взаимодействие с менеджером EXHATARE.EXE реализовано при помощи функций нижнего уровня, а за формирование SCSI-запросов отвечает верхний уровень библиотеки EXHATARE.

Наличие резидентного менеджера EXHATARE.EXE позволяет корректно работать с накопителем EXB-8500 из программ, работающих как в реальном, так и в защищенном режиме процессора i386.

На данный момент пользователи могут получить доступ к устройству EXB-8500 на компьютерах типа IBM PC/AT из программ, написанных на языках высокого уровня Турбо-Паскаль, NDP-C, GNU-C, Borland-C. На базе данного комплекса уже реализован пакет EXAFORT для управления EXB-8500 из программ на ФОРТРАНе [3].

Авторы выражают благодарность И.М. Иванченко и В.В. Коренькову за постановку задачи и полезные обсуждения на всех этапах разработки пакета EXHATARE.

## 7 Индекс

Название функции	Номер	Страница
<b>Нижний уровень</b>		
byte Init_IO_Man (void)	1	6
void AHA_Request (id, buf, blen, cdbptr, cdblen)	2	6
byte AHA_WaitEndOp (id)	3	7
DOSFAR GetDosMem (npara)	4	7
word FreeDosMem (dosptr)	5	8
byte *MapDosMem (dosptr, size)	6	8
<b>Верхний уровень</b>		
byte Init_exatape (void)	7	9
byte End_exatape (void)	8	9
<b>Инициализация чтения/записи</b>		
void SCSI_InitRead (id, buf, lbuf, nblk, sili, fixed)	9	9
void SCSI_InitWrite (id, buf, lbuf, nblk, fixed)	10	10
<b>Выполнение SCSI-запросов</b>		
byte SCSI_WriteFMK (id, nfmk, immed)	11	12
byte SCSI_Erase (id, immed, deldata)	12	13
byte SCSI_Space (id, opcode, nblk)	13	13
byte SCSI_Locate (id, nblk, immed, blktype)	14	14
byte SCSI_Rewind (id, immed)	15	14
byte SCSI_Load (id, immed, opcode)	16	15
byte SCSI_TestReady (id)	17	15
byte SCSI_ReadPosition (id, buf, blktype)	18	15
byte SCSI_ModeSense (id, buf)	19	16
byte SCSI_ModeSelect (id, buf)	20	16
byte SCSI_RequestSense (id, buf, reset)	21	17
byte SCSI_MediumRemoval (id, prevent)	22	17
byte SCSI_Inquiry (id, buf)	23	17
byte SCSI_ReadBlkLimits (id, buf)	24	18
byte SCSI_Reserve (id, pty3, pty3_id)	25	18
byte SCSI_Release (id, pty3, pty3_id)	26	18
<b>Получение диагностики ошибок</b>		
char * SCSI_ErrStat (st)	27	19
char * SCSI_ErrType (buf)	28	19
char * SCSI_ErrMess (buf)	29	20

## Приложение 1. Коды завершения SCSI-запросов, возвращаемые функцией AHA\_WaitEndOp

Код возврата (десятичн.)	Комментарий
0	нет ошибок
2	ошибка передачи данных; ошибка физического чтения или записи; любая нестандартная ситуация
8	устройство выполняет внутреннюю операцию, нужно повторить запрос через некоторое время
24	устройство заблокировано другим инициатором, см. функции SCSI_Reserve и SCSI_Release
66	устройство с заданным SCSI-адресом отсутствует либо выключено питание

## Приложение 2. Описание структуры StdInqStr

```
typedef struct {
    byte DevTypeCode; /* 01- tape, 0x7F - invalid unit*/

    byte DevTypeQual:7; /* 0 for Exabyte (no qualifiers)*/
    byte RMB :1; /* 1 if Media is removable*/

    byte ANSIVersion:3; /* 0 if device isn't ANSI standards*/
    byte ECMAVersion:3; /* 0 if device isn't ECMA standards*/
    byte ISOVersion :2; /* 0 if device isn't ISO standards*/

    byte RespDataFmt:4; /* 2 for Exabyte (ANSI SCSI-2)*/
    byte Reserved1 :2;
    byte TrmIOP :1; /* 1 if Terminate I/O supported*/
    byte AENC :1;

    byte AdditLength; /* Length of additional Inquiry data*/
    /* Additional data starts from the next field*/
    byte Reserved2[2];

    byte SftRe :1; /* soft reset alternative is supported*/
    byte CmdQue :1; /* 1 if tag command queuing is supported*/
    byte Reserved3:1;
    byte Linked:1; /* 1 if linked commands are supported*/
    byte Sync :1; /* synchronous data transfer is supported*/
    byte WBus16 :1; /* device supports 16-bit-wide transfers*/
    byte WBus32 :1; /* device supports 32-bit-wide transfers*/
    byte RelAdr :1; /* device supports relative addressing*/

    char VendorID [ 8]; /* ASCII str. of the vendor name */
    char ProductID [16]; /* ASCII str. of the product name */
    char ProductRev[ 4]; /* ASCII of the product revision */
    char VendorSpec[20]; /* ASCII string with OEM data */
    char Reserved4 [40];
    char SerialNum [10]; /* ASCII string with serial number */
} StdInqStr;
```

## Приложение 3. Описание структуры ReadPosStr

```
typedef struct {
    byte Reserved1:2;
    byte BPU:1; /* 1 if Block Position Unknown */
    /* and this block data is not valid */
    byte Reserved2:3;
    byte EOP:1; /* 1 if tape between LEOT and PEOT*/
    byte BOP:1; /* 1 if tape is at LBOT*/

    byte Partition; /* Always 0*/
    byte Reserved3[2];
    byte FirstBlockLocation[4];
    /* Current logical block position*/
    byte LastBlockLocation[4]; /* not valid, 0 returned*/
    byte Reserved4;
    byte BlocksInBuffer[3]; /* always 0*/
    byte BytesInBuffer[4]; /* always 0*/
} ReadPosStr;
```



## Приложение 4. Описание структуры ReqSenseStr

```
typedef struct {
    byte ErrorCode:7;
    byte Valid :1; /* 1 if "Information" field is valid*/
    byte SegmentNumber; /* Always 0 for Exabyte*/
    byte SenseKey :4; /* see EXB-8500 8mm user's manual*/
    byte Reserved1:1;
    byte ILI:1; /* 1 if the logical block length */
        /* did not match the actual block length on the tape*/
    byte EOM:1; /* 1 LEOT or LBOT = Any End Of Medium*/
    byte FMK:1; /* 1 if the current command detected FMK */

    byte Information[4];
        /* Number of unprocessed blocks or bytes (MSB first)*/
    byte AdditLength;
        /* Additional sense length, excluding this byte */
        /* (15h for exabyte)*/
    byte Reserved2[2];
    byte Reserved3;
    byte OverrunCounter; /* Overrun/Underrun */
        /* (Empty buf on write/Full buf on read) */
    byte AdditCode; /* Additional sense code + */
        /* Additional sense code qualififer */
    byte AdditQual; /* provides additional information */
        /* about each sense key*/
    byte Reserved4[2];
    byte ErrorCounter[3];
        /* In write mode - # of blocks rewritten. */
        /* In read mode # of ECC reconstructed*/

    byte LBOT:1; /* Logical Beginnig Of Tape*/
    byte TNP :1; /* Tape Not present*/
    byte TME :1; /* Tape Motion Error */
    byte ECO :1; /* Error Counter Overflow */
    byte ME :1; /* Media error*/
    byte FPE :1; /* Formatted parity error */
        /* in internal 1Mb data buffer */
    byte BPE :1; /* Bus parity error*/
    byte PF :1; /* Power fail (internal reset on power-up)*/
```

```
byte FE :1; /* Formatter Error (hardware error)*/
byte SSE :1; /* Servo System Error (hardware error)*/
byte WE1 :1; /* Write Error 1 */
        /* (Maximum # of write retries attempted)*/
byte URE :1; /* Under Run Error */
        /* (FE bit will be also set)*/
byte FMKE:1; /* Filemark writing Error */
byte WP :1; /* Write Protect*/
byte TMD :1; /* Tape Mark Detect Error */
        /* (SPACE FileMark not completely performed, */
        /* tape position now not match the requested one) */
byte Reserved5:1;

byte WSEO:1; /* Hardware problem: Write Splice Error */
        /* (Overshot postion when trying to splice).*/
byte WSEB:1; /* Hardware problem: Write Splice Error */
        /* (encountered blank tape).*/
byte PEOT:1; /* Physical End Of Tape*/
byte Reserved6:5;

byte Reserved7;
byte RemainingTape[3]; /* Value in 1K physical blocks */
        /* 0 if no cartridge*/
byte TrRetryCounter; /* Tracking Retry Counter*/
byte RWRetryCounter; /* Read-Write Retry Counter*/
byte FaultSymptom; /* Device specific symptom code.*/
} ReqSenseStr;
```

## Приложение 5. Описание структуры ModeSenseStr

```
typedef struct {
    byte ModeDataLen; /* Total number of ModeSense bytes */
                       /* excluding this one */
    byte MediumType; /* Currently loaded cartridge type*/
    byte Speed:4; /* always 0*/
    byte BufferedMode:3; /* 0- Unbuffered, 1- buffered */
    byte WP:1; /* Cartridge is write-protected */

    byte BlockDescriptorLen;

    byte DensityCode; /* 0- 8500 format, 14h- 8200 format,*/
                       /* 7Fh - no change is desired */
    byte NumberOfBlocks[3]; /* (always 0) */
    byte Reserved1;
    byte BlockLength[3];
        /* 0 - variable len blocks, !=0 - fixed blocks, */
        /* Vendor Unique (we are in the PF=0 format) */
    byte NAL:1; /* 1 - No Auto Load (disable) */
    byte PE :1; /* 1 - Parity Checking Enable */
    byte EBD:1; /* Even Byte Disconnect (1 - Even only) */
    byte NBE:1; /* 1, ignored in 8500 */
    byte Reserved2:1;
    byte ND:1; /* 1 - No Disconnect during data transfer */
    byte Reserved3:1;
    byte CT:1; /* Cartridge type 0- P6, 1- PI (intl) */

    byte P5:1; /* 1- European P5, overrides CT */
    byte Reserved4:7;

    byte MotionThreshold;
        /* in 4KBytes increments, default is 80h,*/
        /* range from 20h to D0h */
    byte ReconnectThreshold; /* ---.--.-- */
    byte GapThreshold; /* range from 0 to 7. */
        /* Indicates number of consecutive gaps before stop */
} ModeSenseStr;
```

## Приложение 6. Описание структуры BlockLimStr

```
typedef struct {
    byte Reserved;
    byte MaxBlockLength[3];
    byte MinBlockLength[2];
} BlockLimStr;
```

## Список литературы

- [1] MS-DOS Bible, The Waite Group's, Howard W.Sams & Company.  
A division of Macmillan, Inc.  
Indianapolis, Indiana, USA, 1988.
- [2] EXB-8500 8mm Cartridge Tape Subsystem, User's Manual.  
EXABYTE Corporation.  
Boulder, Colorado, 1990.
- [3] Кретов Ю.А., Скрипничук А.Г.  
Пакет программ EXAFORT для управления устройством EXB-8500  
на IBM PC/AT из программ на ФОРТРАНе.  
ОИЯИ, P10-94-492, Дубна, 1994.
- [4] Общая библиотека для Си и Паскаля.  
Мир ПК, N1, 1993, стр. 97-104.
- [5] American National Standard for information systems.  
SMALL COMPUTER SYSTEM INTERFACE - 2 (SCSI-2)  
March 9, 1990.  
*Copies of this document may be purchased from:*  
Global Engineering Documents, 2805 McGaw, Irvine, CA 92714.
- [6] АНА-1540а/1542а USER'S MANUAL.  
Adaptec, Inc., 1989.

Рукопись поступила в издательский отдел  
23 декабря 1994 года.